**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2**

**MODUL VII**

**STRUCK & ARRAY**



Oleh:

RIZKULLOH ALPRIYANSAH

2311102142

IF-11-08

**PROGRAM STUDI S1 INFORMATIKA**

**FAKULTAS INFORMATIKA**

**UNIVERSITAS TELKOM PURWOKERTO**

**2024**

1. **DASAR TEORI**
   * + 1. Array

Array adalah struktur data yang menyimpan kumpulan elemen dengan tipe data yang sama, di mana setiap elemen dapat diakses melalui indeks. Array dalam Go memiliki ukuran tetap yang didefinisikan saat deklarasi.

Karakteristik Array di Go:

* Tipe Data Homogen: Semua elemen array harus memiliki tipe data yang sama.
* Ukuran Tetap: Ukuran array ditentukan saat deklarasi dan tidak dapat diubah.
* Indeks Dimulai dari Nol: Elemen pertama berada di indeks 0, dan elemen terakhir berada di indeks panjang array - 1.
* Tipe Data: Array di Go termasuk tipe data statik, sehingga ukuran dan tipe elemen array menjadi bagian dari definisinya.

Deklarasi Array:

Deklarasi array dapat dilakukan dengan cara berikut:

|  |
| --- |
| var namaArray [n]tipeData |

Contoh

|  |
| --- |
| var angka [5]int // Array integer dengan 5 elemen |

Keunggulan Array:

* Struktur data sederhana untuk pengelolaan data homogen.
* Waktu akses cepat karena elemen disimpan secara berurutan di memori.

Keterbatasan Array:

* Ukuran tetap sehingga kurang fleksibel untuk data dinamis.
* Tidak memiliki metode bawaan untuk manipulasi seperti *append* atau *delete* (gunakan slice untuk fleksibilitas).

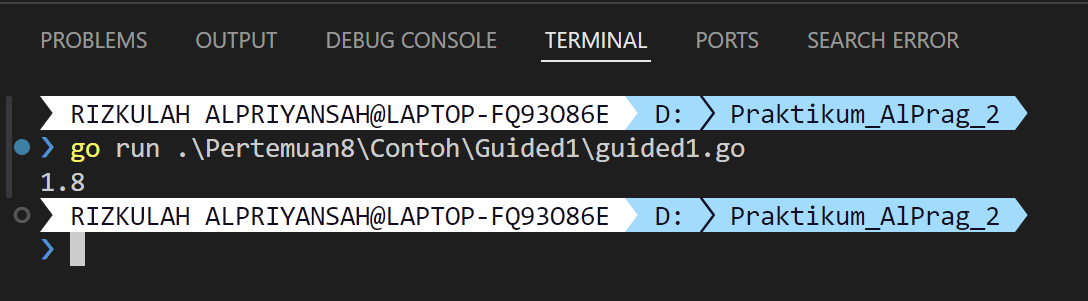
1. **GUIDED**

**Guided1**

**Source Code**

|  |
| --- |
| //Guided 1 - Alias (Type)  package main  import "fmt"  type bilangan int  type pecahan float64  func main() {      var a, b bilangan      var hasil pecahan      a = 9      b = 5      hasil = pecahan(a) / pecahan(b)      fmt.Println(hasil)  } |

**ScreenShot Output**

****

**Deskripsi**

Program di atas mendemonstrasikan penggunaan tipe data buatan (alias) dalam bahasa Go. Tipe bilangan didefinisikan sebagai alias untuk tipe data int, sedangkan pecahan sebagai alias untuk float64. Dalam fungsi main, dua variabel bertipe bilangan dideklarasikan dan diinisialisasi dengan nilai 9 dan 5. Nilai-nilai tersebut kemudian dikonversi ke tipe pecahan untuk melakukan operasi pembagian, yang hasilnya disimpan dalam variabel bertipe pecahan dan ditampilkan menggunakan fungsi fmt.Println.

**Guided2**

**Source Code**

|  |
| --- |
| //Guided 2 - Slice  package main  import (      "fmt"  )  // Fungsi untuk mengecek apakah nama sudah ada di dalam slice  func sudahAda(daftarTeman []string, nama string) bool {      for \_, teman := range daftarTeman {          if teman == nama {              return true          }      }      return false  }  func main() {      // Slice awal untuk daftar teman dengan beberapa data      daftarTeman := []string{"Andi", "Budi", "Cici"}      // Nama-nama baru yang ingin ditambahkan      namaBaru := []string{"Dewi", "Budi", "Eka"}      // Menambahkan nama baru hanya jika belum ada di daftar      for \_, nama := range namaBaru {          if !sudahAda(daftarTeman, nama) {              daftarTeman = append(daftarTeman, nama)          } else {              fmt.Println("Nama", nama, "sudah ada dalam daftar.")          }      }      // Menampilkan daftar teman akhir      fmt.Println("Daftar Teman:", daftarTeman)  } |

**ScreenShot Output**

**A screenshot of a computer program

Description automatically generated**

**Deskripsi**

Program di atas menggunakan konsep slice dalam bahasa Go untuk mengelola daftar teman. Slice awal berisi daftar nama, sementara ada array lain berisi nama-nama baru yang ingin ditambahkan. Sebuah fungsi bernama sudahAda dibuat untuk memeriksa apakah suatu nama sudah ada dalam slice. Pada bagian main, nama-nama baru akan ditambahkan ke slice hanya jika belum ada di dalam daftar. Jika nama sudah ada, program akan mencetak pemberitahuan. Hasil akhirnya adalah daftar teman yang diperbarui ditampilkan menggunakan fungsi fmt.Println.

**Guided3**

**Source Code**

|  |
| --- |
| //Guided 3 - Map  package main  import (      "fmt"  )  func main() {      // Membuat map dengan nama buah sebagai kunci dan harga sebagai nilai      hargaBuah := map[string]int{          "Apel":  5000,          "Pisang": 3000,          "Mangga": 7000,      }      // Menampilkan harga dari setiap buah      fmt.Println("Harga Buah:")      for buah, harga := range hargaBuah {          fmt.Printf("%s: Rp%d\n", buah, harga)      }      fmt.Print("Harga buah Mangga = ", hargaBuah["Mangga"])  } |

**ScreenShot Output**

**A screen shot of a computer

Description automatically generated**

**Deskripsi**

Program di atas mendemonstrasikan penggunaan map dalam bahasa Go untuk menyimpan pasangan data berupa nama buah sebagai kunci (key) dan harga sebagai nilai (value). Map hargaBuah diinisialisasi dengan beberapa data. Dalam fungsi main, program mencetak daftar harga buah dengan menggunakan loop for range, yang iterasi melalui setiap pasangan kunci dan nilai. Selain itu, harga buah tertentu seperti Mangga dapat diakses langsung dengan menyebutkan kuncinya. Hasilnya ditampilkan menggunakan fungsi fmt.Printf dan fmt.Print.

1. **UNGUIDED**

**Unguided1**

**Source Code**

|  |
| --- |
| package main  import (      "fmt"      "math"  )  type Titik struct {      x\_142, y\_142 int  }  type Lingkaran struct {      titikPusat\_142 Titik      radius\_142     int  }  func jarak\_142(a\_142, b\_142 Titik) float64 {      return math.Sqrt(float64((a\_142.x\_142-b\_142.x\_142)\*(a\_142.x\_142-b\_142.x\_142) + (a\_142.y\_142-b\_142.y\_142)\*(a\_142.y\_142-b\_142.y\_142)))  }  func dalamLingkaran\_142(t\_142 Titik, l\_142 Lingkaran) bool {      return jarak\_142(t\_142, l\_142.titikPusat\_142) <= float64(l\_142.radius\_142)  }  func main() {      var cx1\_142, cy1\_142, r1\_142 int      var cx2\_142, cy2\_142, r2\_142 int      var px\_142, py\_142 int      fmt.Println("Koordinat titik pusat & radius lingkaran 1 (cx cy r):")      fmt.Scan(&cx1\_142, &cy1\_142, &r1\_142)      fmt.Println("Koordinat titik pusat & radius lingkaran 2 (cx cy r):")      fmt.Scan(&cx2\_142, &cy2\_142, &r2\_142)      fmt.Println("Koordinat titik sembarang (x y):")      fmt.Scan(&px\_142, &py\_142)      lingkaran1\_142 := Lingkaran{Titik{cx1\_142, cy1\_142}, r1\_142}      lingkaran2\_142 := Lingkaran{Titik{cx2\_142, cy2\_142}, r2\_142}      titik\_142 := Titik{px\_142, py\_142}      inLingkaran1\_142 := dalamLingkaran\_142(titik\_142, lingkaran1\_142)      inLingkaran2\_142 := dalamLingkaran\_142(titik\_142, lingkaran2\_142)      if inLingkaran1\_142 && inLingkaran2\_142 {          fmt.Println("Titik di dalam lingkaran 1 dan 2")      } else if inLingkaran1\_142 {          fmt.Println("Titik di dalam lingkaran 1")      } else if inLingkaran2\_142 {          fmt.Println("Titik di dalam lingkaran 2")      } else {          fmt.Println("Titik di luar lingkaran 1 dan 2")      }  } |

**Screenshot**

**A screenshot of a computer

Description automatically generated**

**Deskripsi**

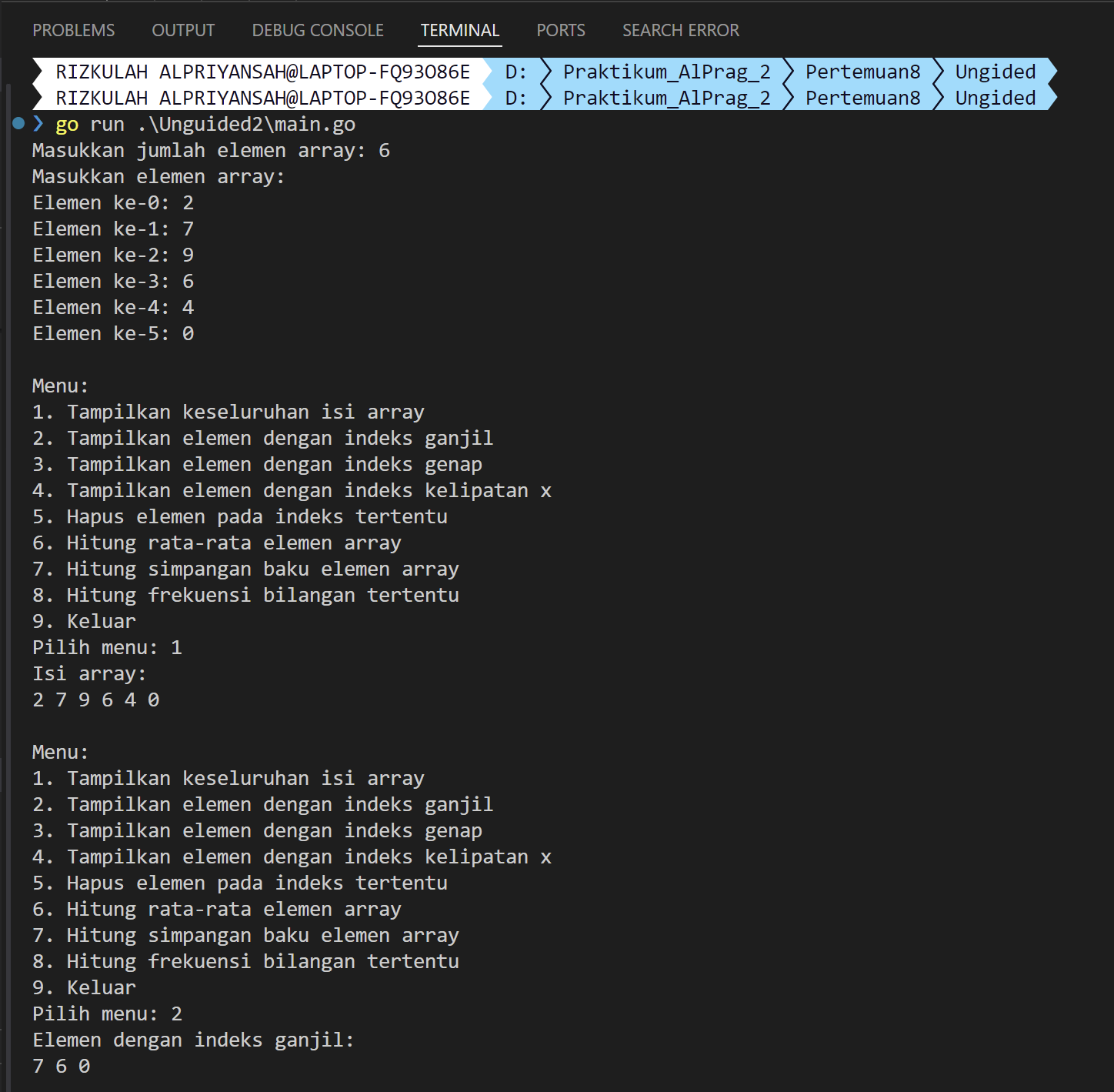
Program di atas menggunakan tipe data struct dalam bahasa Go untuk merepresentasikan koordinat titik dan lingkaran. Struct Titik menyimpan koordinat x dan y, sedangkan struct Lingkaran menyimpan titik pusat dan radius lingkaran. Fungsi jarak\_142 menghitung jarak antara dua titik, sedangkan fungsi dalamLingkaran\_142 memeriksa apakah suatu titik berada di dalam lingkaran berdasarkan jaraknya dari titik pusat. Program membaca input untuk dua lingkaran dan satu titik sembarang, kemudian menentukan apakah titik tersebut berada di dalam salah satu atau kedua lingkaran. Hasilnya ditampilkan melalui konsol.

**Unguided2**

**Source Code**

|  |
| --- |
| package main  import (      "fmt"      "math"  )  func main() {      var n\_142 int      fmt.Print("Masukkan jumlah elemen array: ")      fmt.Scan(&n\_142)      array\_142 := make([]int, n\_142)      fmt.Println("Masukkan elemen array:")      for i\_142 := 0; i\_142 < n\_142; i\_142++ {          fmt.Printf("Elemen ke-%d: ", i\_142)          fmt.Scan(&array\_142[i\_142])      }      for {          fmt.Println("\nMenu:")          fmt.Println("1. Tampilkan keseluruhan isi array")          fmt.Println("2. Tampilkan elemen dengan indeks ganjil")          fmt.Println("3. Tampilkan elemen dengan indeks genap")          fmt.Println("4. Tampilkan elemen dengan indeks kelipatan x")          fmt.Println("5. Hapus elemen pada indeks tertentu")          fmt.Println("6. Hitung rata-rata elemen array")          fmt.Println("7. Hitung simpangan baku elemen array")          fmt.Println("8. Hitung frekuensi bilangan tertentu")          fmt.Println("9. Keluar")          fmt.Print("Pilih menu: ")          var pilihan\_142 int          fmt.Scan(&pilihan\_142)          switch pilihan\_142 {          case 1:              fmt.Println("Isi array:")              for \_, val\_142 := range array\_142 {                  fmt.Print(val\_142, " ")              }              fmt.Println()          case 2:              fmt.Println("Elemen dengan indeks ganjil:")              for i\_142 := 1; i\_142 < len(array\_142); i\_142 += 2 {                  fmt.Print(array\_142[i\_142], " ")              }              fmt.Println()          case 3:              fmt.Println("Elemen dengan indeks genap:")              for i\_142 := 0; i\_142 < len(array\_142); i\_142 += 2 {                  fmt.Print(array\_142[i\_142], " ")              }              fmt.Println()          case 4:              var x\_142 int              fmt.Print("Masukkan nilai x: ")              fmt.Scan(&x\_142)              fmt.Print("Elemen dengan indeks kelipatan ", x\_142, ": ")              if x\_142 > 0 {                  for i\_142 := x\_142; i\_142 < len(array\_142); i\_142 += x\_142 {                      fmt.Print(array\_142[i\_142], " ")                  }              } else {                  fmt.Println("Nilai x harus lebih besar dari 0.")              }              fmt.Println()          case 5:              var idx\_142 int              fmt.Print("Masukkan indeks yang ingin dihapus: ")              fmt.Scan(&idx\_142)              if idx\_142 >= 0 && idx\_142 < len(array\_142) {                  array\_142 = append(array\_142[:idx\_142], array\_142[idx\_142+1:]...)                  fmt.Println("Array setelah penghapusan:")                  for \_, val\_142 := range array\_142 {                      fmt.Print(val\_142, " ")                  }                  fmt.Println()              } else {                  fmt.Println("Indeks tidak valid!")              }          case 6:              sum\_142 := 0              for \_, val\_142 := range array\_142 {                  sum\_142 += val\_142              }              rataRata\_142 := float64(sum\_142) / float64(len(array\_142))              fmt.Printf("Rata-rata elemen array: %.2f\n", rataRata\_142)          case 7:              sum\_142 := 0              for \_, val\_142 := range array\_142 {                  sum\_142 += val\_142              }              mean\_142 := float64(sum\_142) / float64(len(array\_142))              var variance\_142 float64              for \_, val\_142 := range array\_142 {                  variance\_142 += math.Pow(float64(val\_142)-mean\_142, 2)              }              variance\_142 /= float64(len(array\_142))              stdDev\_142 := math.Sqrt(variance\_142)              fmt.Printf("Simpangan baku elemen array: %.2f\n", stdDev\_142)          case 8:              var target\_142 int              fmt.Print("Masukkan bilangan yang ingin dihitung frekuensinya: ")              fmt.Scan(&target\_142)              count\_142 := 0              for \_, val\_142 := range array\_142 {                  if val\_142 == target\_142 {                      count\_142++                  }              }              fmt.Printf("Frekuensi bilangan %d: %d kali\n", target\_142, count\_142)          case 9:              // Keluar dari program              fmt.Println("Keluar dari program.")              return          default:              fmt.Println("Pilihan tidak valid!")          }      }  } |

**Screenshot**

****



A screenshot of a computer

Description automatically generated

**A screenshot of a computer

Description automatically generated**

**Deskripsi**

Program ini adalah aplikasi interaktif dalam bahasa Go (Golang) untuk mengelola array dengan berbagai operasi. Pengguna dapat memasukkan elemen array dan memilih menu untuk menampilkan seluruh isi array, elemen berdasarkan indeks ganjil, genap, atau kelipatan tertentu, serta menghapus elemen pada indeks tertentu. Selain itu, program juga dapat menghitung rata-rata nilai elemen, simpangan baku, dan frekuensi kemunculan suatu bilangan dalam array. Dengan loop tak terbatas, program akan terus berjalan hingga pengguna memilih opsi untuk keluar. Semua operasi dilakukan secara dinamis berdasarkan input dari pengguna.

**Unguided3**

**Source Code**

|  |
| --- |
| package main  import (      "fmt"  )  func main() {      var klubA\_142, klubB\_142 string      var skorA\_142, skorB\_142 int      var hasil\_142 []string      fmt.Print("Klub A: ")      fmt.Scanln(&klubA\_142)      fmt.Print("Klub B: ")      fmt.Scanln(&klubB\_142)      pertandingan\_142 := 1      for {          fmt.Printf("Pertandingan %d : ", pertandingan\_142)          fmt.Scan(&skorA\_142, &skorB\_142)          if skorA\_142 < 0 || skorB\_142 < 0 {              break          }          if skorA\_142 > skorB\_142 {              hasil\_142 = append(hasil\_142, fmt.Sprintf("Hasil %d: %s", pertandingan\_142, klubA\_142))          } else if skorB\_142 > skorA\_142 {              hasil\_142 = append(hasil\_142, fmt.Sprintf("Hasil %d: %s", pertandingan\_142, klubB\_142))          } else {              hasil\_142 = append(hasil\_142, fmt.Sprintf("Hasil %d: Draw", pertandingan\_142))          }          pertandingan\_142++      }      fmt.Println("\nDaftar Hasil Pertandingan:")      for \_, h\_142 := range hasil\_142 {          fmt.Println(h\_142)      }      fmt.Println("Pertandingan selesai")  } |

**Screenshot**

**A screenshot of a computer

Description automatically generated**

**Deskripsi**

Program ini adalah aplikasi interaktif sederhana dalam bahasa Go (Golang) untuk mencatat hasil pertandingan antara dua klub sepak bola. Pengguna memasukkan nama kedua klub, lalu menginput skor untuk setiap pertandingan. Program akan mencatat hasil setiap pertandingan dengan menentukan pemenang berdasarkan skor, atau mencatat "Draw" jika skornya sama. Input akan berhenti jika skor yang dimasukkan negatif. Setelah itu, program menampilkan daftar hasil semua pertandingan yang telah dimainkan dan mengakhiri program dengan pesan penutup.

**Unguided4**

**Source Code**

|  |
| --- |
| package main  import (      "fmt"  )  const NMAX\_142 = 127  type tabel\_142 [NMAX\_142]rune  func isiArray\_142(t\_142 \*tabel\_142, n\_142 \*int) {      \*n\_142 = 0      fmt.Println("Masukkan teks (akhiri dengan '.'): ")      for {          var c\_142 rune          fmt.Scanf("%c", &c\_142)          if c\_142 == '.' {              break          }          t\_142[\*n\_142] = c\_142          (\*n\_142)++      }  }  func cetakArray\_142(t\_142 tabel\_142, n\_142 int) {      for i\_142 := 0; i\_142 < n\_142; i\_142++ {          fmt.Printf("%c", t\_142[i\_142])      }      fmt.Println()  }  func balikkanArray\_142(t\_142 \*tabel\_142, n\_142 int) {      for i\_142, j\_142 := 0, n\_142-1; i\_142 < j\_142; i\_142, j\_142 = i\_142+1, j\_142-1 {          t\_142[i\_142], t\_142[j\_142] = t\_142[j\_142], t\_142[i\_142]      }  }  func palindrom\_142(t\_142 tabel\_142, n\_142 int) bool {      for i\_142 := 0; i\_142 < n\_142/2; i\_142++ {          if t\_142[i\_142] != t\_142[n\_142-i\_142-1] {              return false          }      }      return true  }  func main() {      var tab\_142 tabel\_142      var n\_142 int      isiArray\_142(&tab\_142, &n\_142)      fmt.Print("Teks: ")      cetakArray\_142(tab\_142, n\_142)      balikkanArray\_142(&tab\_142, n\_142)      fmt.Print("Reverse teks: ")      cetakArray\_142(tab\_142, n\_142)      if palindrom\_142(tab\_142, n\_142) {          fmt.Println("Palindrom: True")      } else {          fmt.Println("Palindrom: False")      }  } |

**Screenshot**

A screenshot of a computer

Description automatically generated

**Deskripsi**

Program ini adalah aplikasi dalam bahasa Go (Golang) yang memproses teks berbasis array karakter hingga 127 karakter. Pengguna dapat memasukkan teks secara bertahap, diakhiri dengan karakter `.`. Setelah itu, program mencetak teks asli, membalik urutannya, dan memeriksa apakah teks tersebut merupakan palindrom (teks yang terbaca sama dari depan maupun belakang). Operasi dilakukan menggunakan array statis dan algoritma sederhana untuk pembalikan teks serta deteksi palindrom dengan membandingkan karakter di awal dan akhir array secara berpasangan.

1. **DAFTAR PUSTAKA**

1) Asisten praktikum, Akmelia Zahara dan Kyla Azzahra Kinan “Modul VII STRUCK & ARRAY” Learning Management System, 2024